Rejections Under 35 U.S.C. §102

The Office Action rejected claim 16 under 35 U.S.C. §102(e) as purportedly being unpatentable over Sunkara (6,523,032). Applicant respectfully traverses this rejection.

Sunkara is directed to servicing database requests using multiple database servers. Sunkara teaches that when a number of web/application servers operating in parallel to process traffic require concurrent access to the same database server, the database server creates a bottleneck (Col. 1, lines 46-59). To remedy this, Sunkara discloses using a master database server which contains the entire database, and a plurality of slave database servers which each contains a read-only copy of a portion of the database (Col. 4, lines 46-48). The slave database servers process database read operations, but all write operations are processed by the master database server to prevent consistency problems (Col. 4, lines 49-53). Further, to ensure consistency between the database on the master database server and the copies on the slave database servers, after the master database server performs a write operation, the value written during the write operation is propagated to the slave database servers (Col. 5, lines 28-31).

Claim 16 is directed to a host computer comprising: a processing unit; and a memory interface module to permit accesses by the host computer to a logical entity to be made to a first physical storage location for read requests and to a second physical storage location for write requests, to prevent accesses by the host computer to the logical entity from being made to the second physical storage location for the read requests, and to prevent accesses by the host computer to the logical entity from being made to the first physical storage location for write requests, wherein the first and second physical storage locations are different.

Claim 16 patentably distinguishes over Sunkara because Sunkara fails to disclose or suggest a memory interface module to permit accesses by the host computer to a logical entity to be made to a second physical storage location for write requests, and to prevent accesses by the host computer to the logical entity from being made to the second physical storage location for read requests.

As the Office Action concedes, Sunkara discloses that the master database server can process both read and write operations, while the slave database servers only process read operations. (*see* Page 4, paragraph 5 of Office Action). Specifically, Sunkara discloses that data may be written to the database on the master database server (Col. 4, lines 49-53), and that when

data is not present in one of the copies of the database on the slave database servers, that data may be read from the master database server (Col. 5, lines 62-64). Thus, data may be both read from and written to the master database server.

As seen from the foregoing, contrary to the assertion in the Office Action (page 3), Sunkara does not prevent reads from the master database server, as the master database server of Sunkara may process read and write requests. Thus, Sunkara does not disclose or suggest a module that permits accesses by a host computer to a logical entity to be made to a second physical storage location for write requests, and that prevents accesses by the host computer to the logical entity from being made to the second physical storage location for the read requests.

In view of the foregoing, claim 16 patentably distinguishes over Sunkara. Accordingly, it is respectfully requested that the rejection of claim 16 under 35 U.S.C. §102(e) be withdrawn.

Rejections Under 35 U.S.C. §103

The Office Action rejected claims 1-6, 15, 20, 23-25 and 30 under 35 U.S.C. §103(a) as purportedly being obvious over Sunkara in view Sigal (5,881,292). Applicant respectfully traverses this rejection.

Sigal is directed to a dynamic versioning system for multiple users of multi-module software. Sigal discloses a software system comprised of a master module and a plurality of slave modules (Col. 4, lines 24-27). Sigal further discloses that a plurality of users may request read-only access to the software system, thereby creating a private copy of the requested slave modules for each user in a respective private memory space (Col. 4, lines 55-65). Any modifications made by a user to one of the slave modules is made in the user's private memory space, and is thus known only by the modifying user. However, if a user saves a copy of a modified slave module to the memory space for the software system, a new version of the slave module is created that coexists with the prior versions of the slave module (Col. 4, line 65 – Col. 5, line 19). In this manner, inconsistent changes to the software system performed by different users of the software system may be prevented.

1. The combination of references is improper

The Office Action asserts that "[i]t would have been obvious to one of ordinary skill at the time the invention was made to implement Sigal's method that ensures separate moving of all reads for a logical entity are performed before write updates are moved to the master database with *Sunkara*'s system, because doing so would improve the maintaining of current version applications for each of multiple users without any performance impact." (*See* Page 5, paragraph 7 of Office Action). Applicant respectfully disagrees.

First, Applicant disagrees with the Office Action's characterization of Sigal. The Office Action asserts that Sigal teaches a method of moving a module version from a slave module to a master module. (*See* Page 4, paragraph 6 of Office Action). The Office Action further asserts that Sigal discloses that all writes to the master module are prevented until the last slave module is closed. (*See* Page 5, lines 10-11 of Office Action). The Office Action relies on these assertions as a basis for finding motivation for one of skill in the art to modify the system of Sunkara to include Sigal's method that purportedly "ensures separate moving of all reads for a logical entity are performed before write updates are moved to the master database." (*See* Page 5, paragraph 7 of Office Action).

Applicant has carefully studied the Sigal reference and can find no basis for the assertions in the Office Action that (1) Sigal teaches a method of moving a module version from a slave module to a master module and that (2) Sigal discloses that all writes to the master module are prevented until the last slave module is closed. The Office Action does not cite to any specific portions of Sigal at which these features are believed to be disclosed. If the rejection is to be maintained, Applicant respectfully requests that the Examiner provide citations for the portions of Sigal that are believed to support the rejection.

As Sigal does not disclose that "separate moving of all reads for a logical entity are performed before write updates are moved to the master database," Sigal provides no motivation to modify Sunkara to include this feature.

Thus, it is respectfully asserted that the combination is improper, and that the Office Action fails to establish a *prima facie* case of obviousness of claims 1-6, 15, 20, 23-25, and 30, because there is no motivation in the prior art for the alleged modification of Sunkara. Thus, it is

respectfully requested that the rejection of claims 1-6, 15, 20, 23-25, and 30 under 35 U.S.C. §103 be withdrawn.

2. The claims patentably distinguish over any combination of Sunkara and Sigal

Even if one were to combine Sankara and Sigal, Applicant's claims patentably distinguish over any such combination.

Claim 1

Claim 1 is directed to a method of moving a logical entity from a first storage element to a second storage element, the logical entity being capable of being accessed by a plurality of host computers. The method comprises steps of: creating a copy of the logical entity on the second storage element; moving all reads of the logical entity from each of the host computers to the second storage element; and after the step of moving all reads, moving all writes to the logical entity to the second storage element.

The Office Action asserts that the Sunkara discloses creating a copy of a logical entity on a second storage element (i.e., the master database server), moving all reads of the logical entity to the second storage element, and after the step of moving all reads, moving all writes to the logical entity to the second storage element. (*See* Office Action page 4, paragraph 4). Applicant respectfully disagrees.

First, the Office Action attempts to read the second storage element recited in claim 1 on the master database server of Sunkara. However, the master database server of Sunkara stores the original database. Thus, to the extent that Sunkara can be interpreted to disclose, "creating a copy of the logical entity on the second storage element," any copies of the logical entity are stored on the slave database servers. That is, the master database server cannot be read as the second storage element because a copy of the database is not created on the master database server. The database is built on the master database server, and a copy is not created in the master database of a logical entity already stored on another ("first" in claim 1) storage element.

The slave database servers in Sunkara also cannot be read as the second storage element in claim 1, because Sunkara does not disclose moving all reads and writes to the second storage element. As discussed above, the master database server (i.e., the original location of the logical

entity) may process both read and write requests, but the slave database server may only process read requests. Therefore, all read requests are not moved to the slave database server, as some of the read requests are processed by the master database server. Further, none of the write requests are moved to the slave database server, as all write requests are processed by the master database server. Thus, Sunkara does not disclose or suggests moving all reads or all writes to the slave database server.

As should be clear from the discussion above, Sunkara does not disclose or suggest moving all reads of the logical entity from each of the host computers to a second storage element; and after the step of moving all reads, moving all writes to the logical entity to the second storage element. Thus, claim 1 patentably distinguishes over Sunkara. Accordingly it is respectfully requested that the rejection of claim 1 under 35 U.S.C. §103(a) be withdrawn.

Claims 2-6 and 15 depend from claim 1 and are patentable for at least the same reasons. Accordingly, it is respectfully requested that the rejection of claims 2-6 and 15 under 35 U.S.C. §103(a) be withdrawn.

Claim 20

Claim 20 is directed to a storage management controller for a computer storage system that includes a plurality of storage elements. The storage management controller comprises: an interface module to communicate with the storage elements; and an entity movement manager to control separate moving of a read location and a write location for a specified logical entity.

As should be clear from the discussion above, Sunkara fails to disclose or suggest an entity movement manager to control separate moving of a read location and a write location for a specified logical entity. For example, a write request in the system of Sunkara is always processed by the master database server and is never processed by the slave database servers, Sunkara does not disclose or suggest moving a write location. Therefore, Sunkara fails to disclose or suggest an entity movement manager to control separate moving of a read location and a write location for a specified logical entity.

In view of the foregoing, claim 20 patentably distinguishes over Sunkara. Accordingly, it is respectfully requested that the rejection of claim 20 under 35 U.S.C. §103(a) be withdrawn.

Claim 23

Claim 23 is directed to a computer system, comprising: a plurality of host computers; a plurality of storage elements; and means for separately moving reads for a logical entity and writes for the logical entity from a first physical storage location on one of the storage elements to a second physical storage location on a different one of the storage elements.

As should be clear from the discussion above, Sunkara fails to disclose or suggest means for separately moving reads for a logical entity and writes for the logical entity from a first physical storage location on one of the storage elements to a second physical storage location on a different one of the storage elements. Thus, claim 23 patentably distinguishes over Sunkara. Accordingly, it is respectfully requested that the rejection of claim 23 under 35 U.S.C. §103(a) be withdrawn.